



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Laboratorio di Sicurezza Informatica

Esercitazione: Enumeration e Misconfiguration

Marco Prandini

Andrea Melis

Dipartimento di Informatica – Scienza e Ingegneria

Agenda

■ Enumeration

- Google Dork
- Dns
- Ip
- Subdomains
- Service Enumeration

■ Brute Force

- John
- Hashcat
- PassGen

■ Misconfiguration

- Priv Esc

Google Dork

- Le Google Dorks o “Comandi di Google” costituiscono un metodo per affinare i risultati di ricerca.
- I comandi, anche detti operatori di ricerca, permettono di trovare, attraverso di esse, risultati precisi, corrispondenti al motivo per cui ti sei rivolto a Google.
- Per trovare risultati specifici infatti è sufficiente inserire, all'interno della query, qualche keyword particolare o simbolo in più per trovare risultati specifici.



Dorking as a Cyber Security Tool

- Dorking può essere utile per un pentester nella fase di enumerazione (e non solo) di un determinato target.
- Inoltre, può aiutare a fornire informazioni approfondite quando si tratta di sicurezza e analisi della sicurezza di una architettura di un applicativo Web
- Quindi, in modo legittimo, google (e in generale tanti motori di ricerca che offrono questo tipo di funzionalità) offre uno dei migliori tool di OSINT a disposizione per “mappare” e/o “enumerare” tutte il possibile riguardo ad un preciso target.



Qualche keyword

- **cache**: this dork will show you the cached version of any website, e.g. cache: securitytrails.com
- **allintext**: searches for specific text contained on any web page, e.g. allintext: hacking tools
- **allinurl**: it can be used to fetch results whose URL contains all the specified characters, e.g. allinurl client area
- **filetype**: used to search for any kind of file extensions, for example, if you want to search for jpg files you can use: filetype: jpg
- **inurl**: this is exactly the same as allinurl, but it is only useful for one single keyword, e.g. inurl: admin
- **intitle**: used to search for various keywords inside the title, for example, intitle:security tools will search for titles beginning with “security” but “tools” can be somewhere else in the page.
- **intext**: useful to locate pages that contain certain characters or strings inside their text, e.g. intext:"safe internet"
- **link**: will show the list of web pages that have links to the specified URL, e.g. link: microsoft.com
- **site**: will show you the full list of all indexed URLs for the specified domain and subdomain, e.g. site:securitytrails.com
- *****: wildcard used to search pages that contain “anything” before your word, e.g. how to * a website, will return “how to...” design/create/hack, etc... “a website”.

Esempio: Base

- La struttura generale di una query è la seguente:

"inurl:."domain"/"dorks" "

- Dove:
 - “inurl” = input URL
 - “domain” = dominio che si vuole enumerare
 - “dorks” = dork keyword che si vuole usare



Esempi:

- **Explore LOG Files For Login Credentials**
 - `allintext:password filetype:log after:2020`
- **Dork command using two google operators**
 - `allintext:username filetype:log`
- **Explore Live Cameras**
 - `inurl:top.htm inurl:currenttime`
- **Explore Open FTP Servers**
 - `intitle:"index of" inurl:ftp`
- **Explore last copy cache of a domain**
 - `cache:unibo.it`

Esempi concreti enumerazione target.

- Cerchiamo dei file pdf sul sito ulisse contenente la parola “password”
 - `site:ulisse.unibo.it filetype:PDF intext:password`
- Cerchiamo delle chiavi ssh indicizzate
 - `site:ulisse.unibo.it intitle:index.of id_rsa -id_rsa.pub`

.... Altre idee?



Contromisure

- Tutto ciò che google trova è perché è stato indicizzato.
- Conoscete il file robots.txt?

..... Usatelo! :)

User-agent: *

Disallow: /

Oppure

User-agent: *

Disallow: /cartellachenonvoleteindicizzare

Dns Enumeration

- Per DNS enumeration intendiamo il processo di individuazione di tutti i server DNS e dei record corrispondenti per un determinato target.
- L'elenco dei record DNS fornisce una panoramica del tipo dei record (record di database) registrati del Domain Name System (DNS). Il DNS implementa un database distribuito, gerarchico e ridondante per le informazioni associate ai nomi e agli indirizzi del dominio.



Dns Record Types

dns query	A	Address record , Returns a 32-bit IPv4 address, most commonly used to map hostnames to an IP address of the host, but it is also used for DNSBLs, storing subnet masks in RFC 1101, etc.
dns query	CNAME	Canonical name record , Alias of one name to another: the DNS lookup will continue by retrying the lookup with the new name.
dns query	AAAA	IPv6 address record , Returns a 128-bit IPv6 address, most commonly used to map hostnames to an IP address of the host.
dns query	MX	Mail exchange record , Maps a domain name to a list of message transfer agents for that domain
dns query	NS	Name server record , Delegates a DNS zone to use the given authoritative name servers
dns query	SOA	zone of authority record , Specifies authoritative information about a DNS zone, including the primary name server, the email of the domain administrator, the domain serial number, and several timers relating to refreshing the zone.
dns query	SPF	Sender Policy Framework , a simple email-validation system designed to detect email spoofing by providing a mechanism to allow receiving mail exchangers to check that incoming mail from a domain comes from a host authorized by that domain's administrators.
dns query	TXT	Text record , Originally for arbitrary human-readable text in a DNS record.
dns query	PTR	Pointer record , Pointer to a canonical name. Unlike a CNAME, DNS processing stops and just the name is returned. The most common use is for implementing reverse DNS lookups, but other uses include such things as DNS-SD.
dns query	SRV	Service locator , Generalized service location record, used for newer protocols instead of creating protocol-specific records such as MX.

nslookup

- Nslookup (acronimo di "Name Server Lookup") è un comando utile per ottenere informazioni dal server DNS. È uno strumento di amministrazione di rete per interrogare il Domain Name System (DNS) per ottenere la mappatura del nome del dominio o dell'indirizzo IP o qualsiasi altro record DNS specifico.

nslookup google.com

Server: 127.0.0.1

Address: 127.0.0.1#53

Non-authoritative answer:

Name: google.com

Address: 216.58.208.142

Name: google.com

Address: 2a00:1450:4002:805::200e

nslookup

- Nslookup (acronimo di "Name Server Lookup") è un comando utile per ottenere informazioni dal server DNS. È uno strumento di amministrazione di rete per interrogare il Domain Name System (DNS) per ottenere la mappatura del nome del dominio o dell'indirizzo IP o qualsiasi altro record DNS specifico.

nslookup -type=any google.com

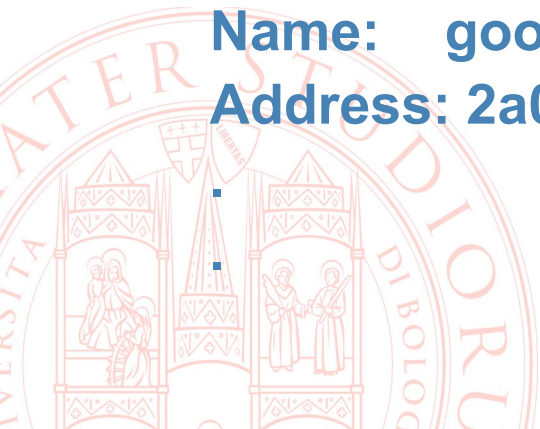
Non-authoritative answer:

Name: google.com

Address: 216.58.208.142

Name: google.com

Address: 2a00:1450:4002:805::200e



Online tools

- <https://centralops.net/> Domain Dossier
- <https://dnsdumpster.com/> Potente suite di Recon
- <https://whois.domaintools.com/> Whois generico.
- <https://reverseip.domaintools.com/> Reverse Lookup
ovvero, partendo da un IP address vengono mostrati
tutti i domini hostati nell'IP specificato, incluso
subdomain.



Altri tool

■ Dnsrecon

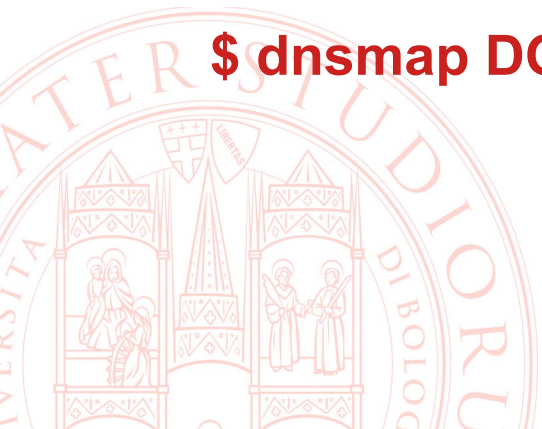
```
# apt install dnsrecon
```

```
$ dnsrecon -d DOMAIN
```

■ Dnsmap

```
# apt install dnsmap
```

```
$ dnsmap DOMAIN
```



Nmap

- **Software open-source per network discovery e security auditing.**
- **Include un potente motore di port scan, con test di vulnerabilità e logiche di discovery incluse di default.**
- **Per range piuttosto grandi con subnet di classe B esistono altri tool più performanti e meno invasivi, come masscan, che aiutano nell'identificazione di servizi e porte aperte ma che non offrono le potenzialità di nmap**



Nmap

- Come prima cosa facciamo un host discovery per scoprire l'ip della macchina vulnerabile sulla rete host-only

- Lanciamo:

nmap --help

Nmap 7.70 (<https://nmap.org>)

Usage: nmap [Scan Type(s)] [Options] {target specification}

- Per fare uno scan sulla rete invece lanciare:

nmap -sn 192.168.56.0/24

■ lista di ip che “rispondono”

Nmap

- A questo punto possiamo usare nmap per fare uno scan delle porte e i servizi disponibili

- Modalità più comune con -A “all”:

nmap -A \$IP

·
·

- Modalità più “invasiva” con salvataggio dell’output si più formati.

nmap -sC -sV -oA output_porte.txt \$IP

·
·

.output porte

Altri tool: Shodan.io

shodan.io/host/137.204.24.147

Apps Security Roba Varia BugBounty Deep MIO CV Studiare Finanza Chimica Didattica Hack Case » Reading list

Shodan Maps Images Monitor Developer More...

SHODAN Explore Pricing Search... Login

137.204.24.147 Regular View Raw Data History

// LAST SEEN: 2022-02-18

General Information

Country	Italy
City	Bologna
Organization	UNI-Bologna
ISP	Consortium GARR
ASN	AS137

Open Ports

80


// 80 / TCP -1427803479 | 2022-02-18T12:02:07.058060

nginx

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Fri, 18 Feb 2022 12:02:06 GMT
Content-Type: text/html
Content-Length: 178
Connection: keep-alive

Subdomain Enumeration

- È una delle parti più cruciali della fase di enumerazione. L'enumerazione dei sottodomini è un processo di ricerca dei sottodomini di uno o più domini root. Secondo RFC 1034, "un dominio è un sottodominio di un altro dominio se è contenuto all'interno di quel dominio".



The diagram illustrates the hierarchical structure of the domain `app.admin.blog.example.com` on a black background with a white zigzag border. The domain is broken down into its constituent parts with labels above and below them:

- `app`: labeled "third-level subdomain" above and "second-level subdomain" below.
- `admin`: labeled "firtst-level subdomain" above and "second-level subdomain" below.
- `blog`: labeled "firtst-level subdomain" above.
- `example`: labeled "top-level domain" above and "root domain" below.
- `com`: labeled "top-level domain" above.

Horizontal lines are placed under `app`, `admin`, `example`, and `com` to indicate the boundaries between the different levels of the domain hierarchy.

Subdomain Enumeration

- L'enumerazione dei sottodomini permette di ampliare enormemente la superficie d'attacco durante una campagna di offensive security.
- L'output tra i più interessanti da trovare in questa fase sono domini nascosti, bloccati o duplicati “minori”.
- Dietro a questo sottodomini possono infatti nascondersi applicativi o servizi “dimenticati”, non aggiornati o non difesi correttamente.
- Una buona strategia di enumerazione sottodomini è quindi fondamentale.



Subdomain Enumeration, precisazioni.

- Un Fully Qualified Domain Name (FQDN) rappresenta un dominio completo per uno specifico host.
- Un FQDN è ad esempio:
 - miopc.ulisse.com ----> Fully Qualified Domain Name
- miopc --> è l'host di ulisse.com (subdomain)
- Però;
 - https://ulisse.com
 - http://myaltropc.ulisse.com
 - https://internal.accounts.ulisse.com
 - https://internal.accounts.pannelloadmin.ulisse.com
- I domini sopra NON sono subdomain di ulisse.com. Sono link a web application (http) hostati sulle porte 80 & 443 dei loro rispettivi host.

Subdomain Enumeration, precisazioni.

- Prendiamo ad esempio `corsosec.ulisse.com`. Dietro questo subdomain potrebbe non esserci nessun applicativo web dietro le classiche porte 80 e 443.
- Questo non significa però che non sia un subdomain valido!
- L'applicativo web infatti potrebbe infatti essere esposto sulla porta 8080, o su un'altra porta.
- Oppure dietro a quel dominio potrebbe essere hostato un altro tipo di servizio non necessariamente di tipo web.
- È quindi buona pratica mappare tutti i subdomain disponibili per un dominio, e risolverli provando a capirne il servizio che c'è dietro.

Subdomain Enumeration.

- Esistono fondamentalmente due strategie:
 - PASSIVA
 - ATTIVA
- Con la strategia Passiva facciamo delle query a dei dataset di DNS noti, che forniscono questi dati, per ottenere tutte le informazioni che cerchiamo, ad esempio
 - Security Trails <https://securitytrails.com/>
 - Shodan <https://www.shodan.io/>
 - Censys <https://censys.io/>
 - Binaryedge <https://www.binaryedge.io/>
 - VirusTotal <https://www.virustotal.com/gui/>
 - ecc

Subdomain Enumeration.

- Esistono molti tool che “wrappano” in automatico tutte le richieste a questi portali noti in un unico output.
- Un esempio molto noto è amass, sviluppato dalla OWASP foundation.
 - <https://github.com/OWASP/Amass>

Altri tool noti

- <https://github.com/projectdiscovery/subfinder>
- <https://github.com/tomnomnom/assetfinder>
- <https://github.com/Findomain/Findomain>



Subdomain Enumeration. CT Abuse

- **Certificate Transparency:** Per poter criptare il traffico per gli utenti, un sito deve innanzitutto richiedere un certificato a un'autorità di certificazione (CA) attendibile.
- Il certificato viene poi presentato al browser per l'autenticazione del sito a cui l'utente desidera accedere. Negli ultimi anni, le CA e i certificati emessi si sono rivelati vulnerabili alla compromissione e alla manipolazione, a causa di falle strutturali nel sistema dei certificati HTTPS.
- Il progetto Certificate Transparency di Google è stato ideato per proteggere il processo di emissione dei certificati offrendo un framework aperto per il monitoraggio e il controllo dei certificati HTTPS.




Subdomain Enumeration. CT Abuse

- Un log di Certificate Transparency è un server che implementa RFC 6962 e consente a qualsiasi parte interessata di inviare certificati che sono stati emessi da un'autorità di certificazione pubblicamente attendibile. Una volta che un log accetta un certificato, le proprietà crittografiche del log assicurano che la voce non potrà mai essere rimossa o modificata.
- Ciò significa che tutti i certificati emessi dalla CA verrebbero aggiunti a un elenco pubblico comune. Avere un registro trasparente di tutti i certificati emessi è un'ottima soluzione per risolvere il problema dell'emissione fraudolenta di certificati, poiché i legittimi proprietari di domini hanno la possibilità di individuare i certificati emessi senza il loro consenso.



Subdomain Enumeration. CT Abuse

- Poiché ogni volta che un'organizzazione ottiene un certificato SSL viene registrata in questi registri CT, è possibile abusarne facilmente. Poiché chiunque può interrogarli, possono essere utilizzati per enumerare i sottodomini di un dominio principale che hanno un certificato TLS associato.
- Possiamo trovare tutti i certificati SSL appartenenti a un dominio inviando una richiesta GET a **<https://crt.sh/?q=%25.dell.com>**

crt.sh Identity Search  Group by Issuer									
Criteria Type: Identity Match: ILIKE Search: 'dell.com'									
Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name		
	2398054168	2020-01-29	2015-07-15	2017-06-12	RUBIN.MARK.ISADORE.1043574880	mark_rubin@federal.dell.com	C=US, O=U.S. Government, OU=DoD, OU=PKI, CN=DOD EMAIL CA-32		
	2398051248	2020-01-29	2012-08-09	2015-08-08	RUBIN.MARK.ISADORE.1043574880	mark_rubin@federal.dell.com	C=US, O=U.S. Government, OU=DoD, OU=PKI, CN=DOD EMAIL CA-30		
	2398012835	2020-01-29	2013-10-23	2014-10-23	Hurley.Brendan.FORC1000038623.Encrypt	Brendan_Hurley@federal.dell.com	C=US, O=U.S. Government, OU=ECA, OU=Certification Authorities, CN=ORC ECA HW.4		
	2398012604	2020-01-29	2013-02-14	2014-02-14	Kamanda.Ernestine.O.ORG1000032582.Encrypt	Ernestine_Kamanda@federal.dell.com	C=US, O=U.S. Government, OU=ECA, OU=Certification Authorities, CN=ORC ECA HW.4		
	2382777596	2020-01-27	2015-04-08	2017-04-08	nexusapp.us.dell.com	nexusapp.us.dell.com	C=NL, L=Amsterdam, O=Verizon Enterprise Solutions, OU=Verizon Public SureServer CA G14-SHA2		
	2382769749	2020-01-27	2015-05-11	2017-05-11	ebi.us.dell.com	ebi.us.dell.com	C=NL, L=Amsterdam, O=Verizon Enterprise Solutions, OU=Verizon Public SureServer CA G14-SHA2		
	2382733309	2020-01-27	2015-07-17	2017-07-17	tpepwucsba01.tpe.apac.dell.com	tpepwucsba01.tpe.apac.dell.com	C=NL, L=Amsterdam, O=Verizon Enterprise Solutions, OU=Verizon Public SureServer CA G14-SHA2		
	2382711925	2020-01-27	2016-06-08	2017-06-08	indiacp.portal.dell.com	indiacp.portal.dell.com	C=US, O="Entrust, Inc.", OU=See www.entrust.net/legal-terms, OU="(c) 2012 Entrust, Inc. - for authorized use only", CN=Entrust Certification Authority - L1K		
	2382711981	2020-01-27	2016-11-08	2017-10-31	dellidm.dell.com	dellidm.dell.com	C=US, O="Entrust, Inc.", OU=See www.entrust.net/legal-terms, OU="(c) 2012 Entrust, Inc. - for authorized use only", CN=Entrust Certification Authority - L1K		
	2382711642	2020-01-27	2016-11-03	2017-11-03	pvwa.us.dell.com	auspwwweb01.aus.amer.dell.com auspwwweb02.aus.amer.dell.com auspwwweb02.us.dell.com pvwa.aus.amer.dell.com pvwa.ins.dell.com pvwa.us.dell.com	C=US, O="Entrust, Inc.", OU=See www.entrust.net/legal-terms, OU="(c) 2012 Entrust, Inc. - for authorized use only", CN=Entrust Certification Authority - L1K		
	2382705131	2020-01-27	2015-01-21	2017-01-21	lync-intowa-blr.dell.com	blrpwlyncowa101.blr.amer.dell.com blrpwlyncowa102.blr.amer.dell.com lync-extowa-blr.dell.com lync-intowa-blr.dell.com	C=NL, L=Amsterdam, O=Verizon Enterprise Solutions, OU=Verizon Public SureServer CA G14-SHA2		
	2382688313	2020-01-27	2014-11-06	2016-11-06	penpwlyncfe107.pen.apac.dell.com	admin.dell.com dialin.dell.com Lyncdiscover.dell.com LyncdiscoverInternal.dell.com meet.dell.com penpwlyncfe107.pen.apac.dell.com webext-penpwlyncfe107.dell.com	C=NL, L=Amsterdam, O=Verizon Enterprise Solutions, OU=Verizon Public SureServer CA G14-SHA2		

Subdomain Enumeration.

- Non c'è una vera e propria difesa da questo tipo di enumerazione, dal momento che i subdomain sono in ogni caso esposti.
- È bene però sapere la propria superficie d'attacco esposta, e a cosa ogni subdomain è associato.

Altre tecniche includono:

- DNS bruteforcing (non fatelo! :))
- Permutation/Alternations
- VHOST probing
- Recursive Enumeration



Brute Force, Hash Crack

- Password cracking è un processo di recupero delle password, mediante l'utilizzo di informazioni che sono state trasmesse da un sistema informatico.
- Un approccio comune (metodo forza bruta, o attacco brute force) è quello di tentare tutte le possibili combinazioni di caratteri, e di confrontarle con un hash crittografico della password
- Un altro possibile approccio è il cosiddetto attacco a dizionario; in questo caso viene generato o recuperato precedentemente un insieme di tutte le possibili soluzioni, da eseguire prima di un attacco a forza bruta



Wordlists

- La wordlist è un insieme di entry che rappresentano le possibili combinazioni che volete testare.
- Le wordlist più comuni sono ovviamente quelle composte dalle password più comuni, più diffuse oppure create ad-hoc in base al vostro target
- Le wordlist però possono pure rappresentare:
 - Lista delle entry di un possibile sito web
 - Lista possibili utenti
 - Lista sottodomini
 - Lista file comuni di un sito web
 - Lista nodi api
 - Ecc.



Wordlists, le più famose

- Ci sono alcuni repository famosi che contengono queste tipologie di wordlist, il più famoso è probabilmente SecList

<https://github.com/danielmiessler/SecLists>

git clone <https://github.com/danielmiessler/SecLists>.git

- Altra wordlist, solo password, molto famosa è rockyou che ormai è diventata molto grande :).

<https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt>



Wordlists Generate

- Le wordlist però è possibile anche generarle. Ci sono diversi metodi e criteri.
- Si possono generare partendo da un testo o una pagina web come fa il software cewl (che vedremo in avanti sulla parte web).
- Si possono generare a partire da delle keyword e generando tutte le possibili permutazioni.
- Ci sono anche tool che combinano questo approccio basandosi su informazioni fornite dall'utente.



Wordlists Generate: cupp

- Cupp è un tool che vi permette generare, ingrandire o migliorare wordlist per attacchi brute-force
- La sua particolarità consiste nell'avere un'interfaccia iterativa che vi permette di inserire dati e keyword a seconda di domande prestabilite e generare la password
- Sulla vm del lab è già installato ed è sufficiente lanciarlo con:

cupp -i

- L'opzione -i è la modalità interattiva



John the Ripper

- John the Ripper è uno strumento di controllo della sicurezza e recupero password
- È un progetto Open Source disponibile per molti sistemi operativi.
- La version di John the Ripper “jumbo” supporta centinaia di tipi di hash e cifratura, tra cui:
 - password utente di versioni Unix (Linux, * BSD, Solaris, AIX, QNX, ecc.),
 - MacOS, Windows,
 - "app web" (ad es. WordPress),
 - groupware (ad esempio, Notes / Domino) e server di database (SQL, LDAP, ecc.);
 - acquisizioni del traffico di rete (autenticazione di rete Windows, WiFi WPA-PSK, ecc.);
 - chiavi private crittografate (SSH, GnuPG, portafogli di criptovaluta, ecc.);
 - ecc

John the Ripper

- Nella VM del corso john è già installato nel path di root, quindi basta lanciare

sudo john

- Installazione alternativa più completa

sudo apt install zlib1g-dev libssl-dev

git clone "https://github.com/magnumripper/JohnTheRipper.git"

cd JohnTheRipper/src

./configure --without-openssl

sudo make -s clean

sudo make -sj4

- Posizionarsi nella cartella “run” e da lì avete tutti gli eseguibili di john, tra cui il principale ovvero:

./john --test

John the Ripper, primo utilizzo

- Come prima cosa lanciamo john nel modo più semplice possibile, da root, sul file etc shadow contenente gli hash – salted dei nostri account della VM del laboratorio.

`./john /etc/shadow`

- Cosa notate? È riuscito a recuperare alcune password? Vediamo lo stato di avanzamento con

`./john /etc/shadow --show`



John the Ripper, primo utilizzo

- John è riuscito a recuperare le password perché in modalità 'Single Crack' Mode:

This is the mode you should start cracking with. It will use the login names, "GECOS" / "Full Name" fields, and users' home directory names as candidate passwords, also with a large set of mangling rules applied.

Since the information is only used against passwords for the accounts it was taken from (and against password hashes which happened to be assigned the same salt), "single crack" mode is much faster than wordlist mode.



John the Ripper con wordlist

- Usiamo ora la funzionalità con wordlist di john. Questo significa che utilizzeremo una wordlist nota per cracckare l'hash di un utente.
- Come prima cosa aggiungiamo quindi un utente con una password “nota”
- Aggiungiamo l'utente con:
`adduser user_test`
- Cambiamone la password con:
`passwd user_test`
New Password: `batman`
Repeat Password: `batman`

Generiamo l'hash delle password

- Con il comando unshadow della suite di John possiamo creare un file, basato sui file passwd e shadow che possa essere letto da John
- Il processo in questione serve se gli hash delle password sono memorizzate in shadow.
- Sempre dalla cartella “run” di John lanciate
`./unshadow /etc/passwd /etc/shadow > brute.txt`



Generiamo l'hash delle password

- Possiamo quindi lanciare john specificando una wordlist

`./john --wordlist=PERCORSO_WORDLIST brute.txt`

- Che differenza c'è se non specifichiamo la wordlist? Diversi risultati? Perché?

`./john brute.txt`



Altri usi

■ Con John è possibile

- Fare (provare) il crack di uno zip
- Di diversi formati hash
 - Md5
 - Sha*
 - NTLM
 - ..
- Brute force su kerberos per:
 - Golden Ticket Attack
 - ReRoast Attack
 - Ecc
- Crack di pdf protetti da password



Hashcat

- Altro tool interessante è hashcat. Funziona secondo lo stesso principio di john.
- Hashcat è probabilmente un tool più funzionale, offre un supporto migliore per la GPU e col passare del tempo anche il generatore di word è migliorato notevolmente.
- Nella macchina del lab è già installato, potete fare dei test anche con hashcat

hashcat --help



Misconfiguration

- In questa parte del lab vedremo alcuni esempi di misconfiguration
- Per misconfiguration intendiamo i casi nei quali una errata configurazione permette ad un utente malintenzionato di poter effettuare un attacco con lo scopo di:
 - Ottenere informazioni sensibili
 - DoS di un servizio
 - Ottenere accesso privilegiato o di un utente di “livello” superiore a quello attuale
 - ecc



Misconfiguration

- Immaginiamo quindi di aver avuto accesso ad una macchina attraverso delle credenziali ottenute tramite brute force.
- Abbiamo accesso ad una macchina, da utente NON privilegiato.
- Vogliamo però avere il controllo della macchina, che strategie abbiamo?

Migliaia.....

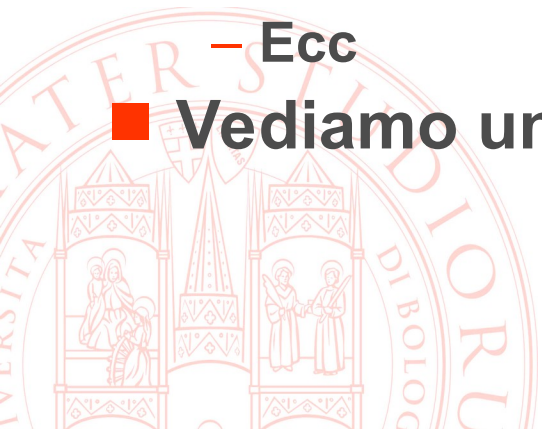


Misconfiguration

- Una delle tante strategie è quella di guardare processi e configurazioni che permettono operazioni (magari singole) privilegiate.
- In questo lab vedremo due esempi di misconfiguration, tipiche di challenge boot2root, che sfrutteremo per fare la cosiddetta *PRIVILEGE ESCALATION*
- Per privilege escalation (o privesc) intendiamo il processo di ottenere i privilegi di un utente “superiore” attraverso vari meccanismi quali:
 - Credenziali
 - Shell utente
 - Esecuzione comandi
 - ecc

Misconfiguration Primo Esempio: sudoers

- Un esempio molto comune è la configurazione del file sudoers
- Come sappiamo, attraverso il file sudoers, è possibile concedere dei privilegi anche molto specifici a particolare utenti, ad esempio:
 - Concedere all'utente www-data di poter modificare i file del sito web hostato dal server
 - Concedere un particolare comando di root ad un utente low level
 - Ecc
- Vediamo un esempio!



Misconfiguration Primo Esempio: sudoers

- Modifichiamo il file sudoers e inseriamo una regola per l'utente user_test creato precedentemente
- Con l'utility visudo, inseriamo la seguente riga
visudo /etc/sudoers

user_test ALL=(root) NOPASSWD: /usr/bin/vi /var/www/html/*

Che significa?

- Possiamo vedere cosa possiamo eseguire come utente privilegiato da un utente NON privilegiato con:

sudo -l

Misconfiguration Primo Esempio: sudoers

- Possiamo sfruttare questa vulnerabilità in almeno 3 modi diversi.

- Primo: Apriamo una shell da vi

```
sudo /usr/bin/vi /var/www/html/file_a_caso  
:!bash
```

- Secondo: Apriamo il file /etc/passwd

```
sudo /usr/bin/vi /var/www/../../etc/passwd  
inseriamo una nuova entry (stessa cosa per /etc/shadow)
```

- Terzo: Modifichiamo uno script che viene eseguito da root!



Misconfiguration Secondo Esempio: SUID

- Un altro esempio molto comune è guardare i file di root con settato il SUID
- Se il bit SUID di un binario è settato significa che è possibile eseguire il binario con privilegi di root
- Se troviamo un binario di questo tipo che possiamo sfruttare per generare una vulnerabilità possiamo effettuare, anche qui, una privilege escalation.



Misconfiguration Secondo Esempio: SUID

- Partiamo da un caso semplice ovvero immaginiamo di avere il suid settato sul binario per copiare file cp
- È possibile settare il suid col comando

```
chmod u+s /bin/cp
```

```
ls -al /bin/cp
```

```
-rwsr-xr-x 1 root root 146880 Feb 28 2019 /bin/cp
```



Misconfiguration Secondo Esempio: SUID

- Che fare? Anche qui le strategie sono diverse.
- L'idea in questo caso è avere un tool che vi permette di copiare (sovrascrivere?) file di root.
- Una possibile soluzione è ad esempio riscrivere il file passwd aggiungendo una nuova entry.



Misconfiguration Secondo Esempio: SUID

- Da utente `user_test` copiamo il `/etc/passwd` da qualche parte, ad esempio su `/tmp`

```
cat /etc/passwd > /tmp/passwd
```

- Dopo di che aggiungiamo una nuova entry al file `passwd`.

- Generiamola prima con `openssl`

```
openssl passwd -1 -salt seclab seclab > new_entry
```

- Modifichiamo l'entry in modo tale che sia un formato compatibile con il file `/etc/passwd`

```
seclab:CONTENUTO_DI_NEW_ENTRY:0:0:/root:/bin/bash
```

- Aggiungiamo la entry al nostro file temporaneo

```
cat new_entry >> /tmp/passwd
```



Misconfiguration Secondo Esempio: SUID

- A questo punto possiamo sovrascrivere il file etc passwd con il nostro file con l'entry aggiuntiva!
Perchè?
Perchè abbiamo il binario cp con suid attivo!

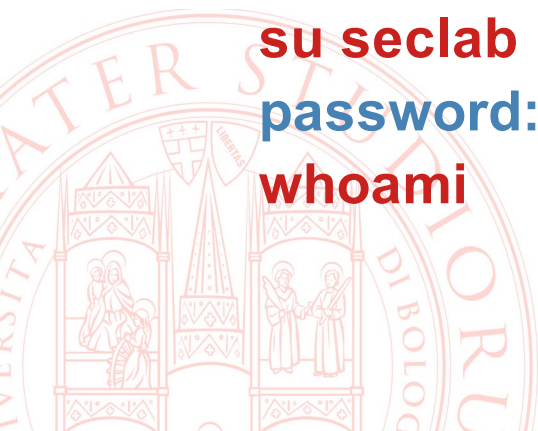
cp /tmp/passwd /etc/passwd

- Possiamo ora loggarci con utente seclab (password seclab) che è un utente con privilegi di root!

su seclab

password: seclab

whoami



Esercitazioni e Materiale Approfondimento

Google Dork

- <https://www.oakton.edu/user/2/rjtaylor/cis101/Google%20Hacking%20101.pdf>
- <https://www.exploit-db.com/google-hacking-database>
- https://www.researchgate.net/publication/335383170_Google_Dorks_-_Advance_Se
- <http://pdf.textfiles.com/security/googlehackers.pdf> (ottimo!)

DNS Enum

- <https://pentestwiki.org/enumeration-cheat-sheet/>
- <https://www.exploit-db.com/docs/english/13687-how-to-dns-enumeration.pdf>

Subdomain Enum

- <https://pentester.land/cheatsheets/2018/11/14/subdomains-enumeration-cheatsheet.html>
- <https://sidxparab.gitbook.io/subdomain-enumeration-guide/>

Nmap

- <https://nmap.org/book/cover/nns-cover.pdf>

Extras: Esercitazioni e Link utili

John e password crack

- <https://www.openwall.com/john/>
- <https://miloserdov.org/?p=4961>
- <https://dfir.science/2014/07/how-to-cracking-zip-and-rar-protected.html>

Challenges a tema (o con anche esercizi a tema) software security

- <https://overthewire.org/wargames/>
- <https://crackmes.one/>
- <https://www.hackthebox.eu/>
- <https://pentesterlab.com/>
- [pwnable.*](https://pwnable.org/)
- <https://github.com/apsdehal/awesome-ctf#wargames>

Link utili

- <https://gtfobins.github.io/>
- <https://github.com/longld/peda>
- [pwntools](https://github.com/longld/pwntools)